

## PENGENDALIAN MESIN *HOIST HANGER* DALAM PROSES PTC/ED MENGGUNAKAN PLC OMRON

Wahyudi \*, M. Hasim As'ari \*\*)

### Abstract

*The development of technology demands efficient and work speed, so it needs a system which can handle several things. This problem can be covered with PLC (Programmable Logic Control) that can integrate several self standing component to be an integrated control system and can be changed the configuration without change all instrument. Modify the ladder program can alter system of PLC. This paper shows Hoist Hanger system in PTC/ED (Pre Treatment Chemical/Electrocoat Disposition) process that controlled by PLC. Above input from limit switch, sensor, or the other input, PLC system drives the motor relay Hoist Hanger machine. That cause the motor relay Hoist Hanger change its condition, so the moving of Hoist Hanger machine will be suitable with PLC program.*

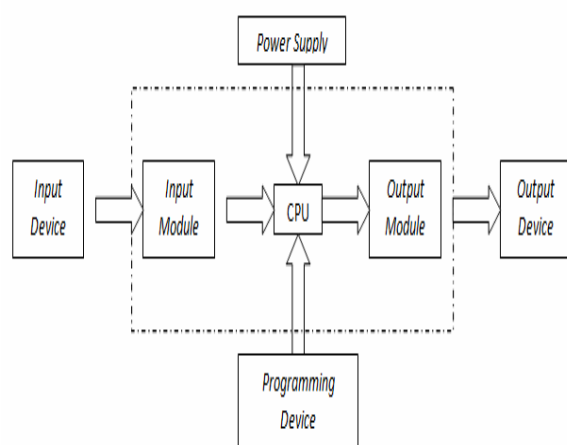
*Key words : PLC Omron, Hoist Hanger*

### Pendahuluan

Dewasa ini perkembangan ilmu pengetahuan dan teknologi semakin pesat, maka tidak dapat dipungkiri bahwa elektronika memegang peranan yang sangat penting dalam dunia industri, terutama dalam proses pengendalian mesin-mesin produksi dan peralatan-peralatan berat secara otomatis. Kemajuan teknologi di bidang elektronika ini menjadi suatu tantangan bagi dunia industri. Efisiensi produksi umumnya dianggap sebagai kunci sukses perusahaan. Oleh karena itu, untuk mengendalikan proses produksi yang berkesinambungan dengan kualitas produk yang terjamin serta memiliki daya saing yang tinggi dengan industri lainnya, diperlukan mesin-mesin berteknologi tinggi dengan sistem pengendalian otomatis. Salah satu pengendali otomatis tersebut adalah *Programmable Logic Control* (PLC).

### Prinsip Kerja PLC

Pada prinsipnya sebuah PLC melalui modul *input* bekerja menerima data-data berupa sinyal dari *external input device* pada sistem yang dikontrol. Prinsip kerja PLC ditunjukkan oleh Gambar 1.



Gambar 1 Diagram blok sistem PLC.

Peralatan *input* luar tersebut antara lain berupa sakelar, tombol, sensor. Data-data masukan yang masih berupa sinyal analog akan diubah oleh modul *input A/D (Analog to Digital input module)* menjadi sinyal digital. Selanjutnya oleh prosesor pusat (CPU) yang ada di dalam PLC, sinyal digital itu diolah sesuai dengan program yang telah dibuat dan disimpan di dalam memori. Kemudian CPU mengambil keputusan dan memberikan perintah melalui modul *output* dalam bentuk sinyal digital. Modul *output D/A (Digital to Analog module)* dari sistem yang dikontrol, antara lain berupa kontaktor, *relay*, *solenoid*, alarm, dimana nantinya dapat mengoperasikan sistem proses yang dikontrol tersebut secara otomatis.

### Diagram Ladder

Diagram *ladder* (tangga) adalah skema penyajian proses kontrol sekuensial. Dalam penggambaran diagram tangga dikenal simbol-simbol sebagai berikut:

#### a. Saklar *Normally Open* (NO)

Saklar ini menandakan keadaan saklar yang normalnya pada posisi Off (terbuka), dan akan On (terhubung) bila *relay* telah *terenergize*. Simbol NO ditunjukkan oleh Gambar 2.

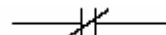


Gambar 2 Simbol NO

Pada Gambar 2, jika saklar *terenergize*, maka kontaktor akan berubah kondisi menjadi tertutup.

#### b. Saklar *Normally Close* (NC)

Saklar ini menandakan keadaan saklar yang normal pada keadaan On (terhubung), jadi jika saklar tersebut diaktifkan akan menjadi Off (terbuka). Simbol NC ditunjukkan oleh Gambar 3.



Gambar 3 Simbol NC

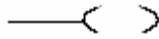
Pada Gambar 3, jika saklar *terenergize*, maka kontaktor akan berubah kondisi menjadi terbuka.

#### c. Keluaran

Keluaran *relay* akan mengaktifkan kontak-kontak NO dan NC. Simbol keluaran ditunjukkan oleh Gambar 4.

\*) Staf Pengajar Jurusan Teknik Elektro Fakultas Teknik Undip

\*\*) Mahasiswa Jurusan Teknik Elektro Fakultas Teknik Undip



Gambar 4 Keluaran *relay*

Keluaran akan aktif jika ada perubahan pada *input* keluaran tersebut.

#### Komponen pada PLC

Komponen pada PLC terdiri dari *latch*, *timer*, *counter*, dan *MCR*. Komponen-komponen tersebut memiliki fungsi yang berbeda. Komponen *latch* berfungsi sebagai penahan, *timer* berfungsi sebagai menunda waktu, *counter* berfungsi sebagai pencacah, dan *MCR* berfungsi mengontrol program.

##### a. Komponen *Latch* (Pengunci)

Komponen ini berfungsi menahan keluaran untuk masukan sesaat. Ada dua jenis fungsi yang berkaitan dengan komponen ini

SET: menahan keluaran untuk status ON (*latch*).

RST: menahan keluaran untuk status OFF (*unlatch*).

##### b. Komponen *Timer*

Ada tiga macam jenis komponen *timer* yaitu:

TON: ON *delay timer*, yaitu menunda waktu hidup selama selang waktu tertentu.

TON: OFF *delay timer*, yaitu menunda waktu mati selama selang waktu tertentu.

TMR: *integrating timer*, yaitu menunda waktu hidup selama selang integral waktu tertentu.

##### c. Komponen *Counter*

Beberapa fungsi yang berkaitan dengan *counter* diantaranya adalah:

CTU: *Up counter*

Nilai *counter* akan dinaikkan untuk setiap pulsa yang masuk dan kontaktor akan berubah kondisinya jika pulsa yang masuk telah sama dengan nilai *setting* pada *counter* tersebut.

CTD: *Down counter*

Kebalikan dari CTU, nilai *counter* akan diturunkan, kontaktor akan berubah kondisi jika pulsa yang masuk telah sama dengan nilai *setting* pada *counter* tersebut.

CTUD: *Up/Down counter*

Merupakan gabungan dari CTU dan CTD serta *reset*.

#### Pemrograman PLC

Saat ini fasilitas PLC dengan komputer sangat penting artinya dalam pemrograman PLC dalam dunia industri. Sekali sistem diperbaiki, program yang benar dan sesuai harus disimpan ke dalam PLC lagi. Selain itu perlu dilakukan pemeriksaan program PLC, apakah selama disimpan tidak terjadi perubahan atau sebaliknya, apakah program sudah berjalan dengan benar tidak.

##### a. Langkah-Langkah Pemrograman PLC

Sebelum menuliskan program pada PLC hendaknya kita mengetahui sistematika disain terlebih dahulu.

#### Langkah-Langkah Dasar Pemrograman :

1. Mengetahui cara kerja sistem yang akan dikontrol.
  2. Memberikan *output* dan *input* ke I/O bit, merancang peralatan luar yang akan mengirim dan menerima sinyal dari PLC.
  3. Menggunakan simbol *Ladder*, gambar diagram untuk menampilkan sekuensial dari operasinya.
  4. Jika menggunakan *Programming Console*, mengkodekan simbol *Ladder* ke dalam daftar instruksi *mnemonic*, sehingga program dapat dimasukkan ke CPU melalui *Programming Console*.
  5. Mengecek kesalahan program.
  6. Membetulkan kesalahan program dengan merubah program.
  7. Menjalankan program dengan mencobanya.
  8. Bila terjadi kesalahan, membetulkan kesalahan program dengan merubah program.
- b. Bahasa Program

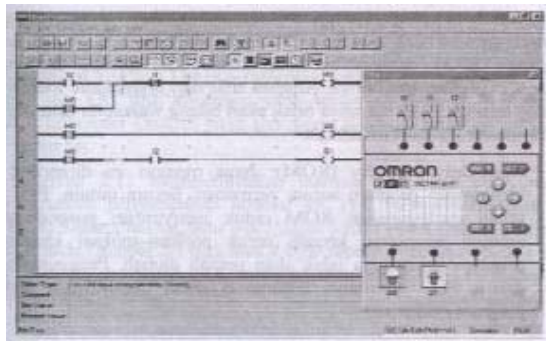
Kontroler PLC dapat diprogram melalui komputer, tetapi juga bisa diprogram melalui program manual, yang biasa disebut dengan konsol (*console*). Untuk keperluan ini, dibutuhkan perangkat lunak yang biasanya tergantung pada produk PLC. Dengan kata lain, masing-masing produk PLC membutuhkan perangkat sendiri-sendiri. *Programming console* ditunjukkan oleh Gambar 5.



Gambar 5 *Programming Console*.

*Programming console* biasanya digunakan untuk program PLC yang sederhana, yaitu dengan cara mengetikkan kode-kode pada *console* tersebut.

Untuk memprogram PLC menggunakan PC, diperlukan *software* yang sesuai dengan PLC yang diprogram, dimana untuk setiap vendor yang berbeda menggunakan *software* yang berbeda pula. Bahasa program yang digunakan sudah dikonversi menjadi bahasa yang dimengerti manusia. Khususnya memakai istilah, simbol, dan gambar teknik standar yang sudah dikenal. Bahasa program disajikan dalam dua bentuk yaitu diagram tangga (*Ladder Diagram*) dan tabel *Mnemonic*. Tampilan program menggunakan PLC ZEN Omron ditunjukkan oleh Gambar 6.



Gambar 6 Tampilan program ZEN Omron.

Pemrograman PLC menggunakan ZEN Omron, yaitu dengan cara membuat diagram *ladder* pada program tersebut. Untuk mengetahui program yang dibuat dapat berjalan atau tidak, maka harus disimulasikan. Program yang dapat berjalan ditandai dengan nyala lampu dari *simulator* program.

### Perancangan Sistem PTC/ED

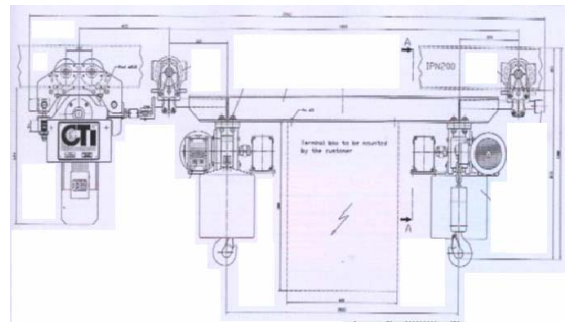
Pengecatan warna dasar *body* dilakukan di sebuah *plant* yang disebut *monorail*. *Monorail* adalah rangkaian seperti kereta gantung yang disertai dengan alat pengangkat *body* (*hoist*) yang kemudian mengangkat dan membawa *body* untuk dicelupkan pada bak-bak yang berisi air, pembersih dan cat. Pada sistem *monorail* terjadi proses PTC/ED (*Pre Treatment Chemical/ Electrocoat Deposition*) *Monorail* ini dikendalikan oleh PLC (*Programmable Logic Controller*). Tiap bak disebut *stage*, berikut adalah *stage-stage monorail* :

- *Stage 0 : Preparation*  
Pada tahap ini *body* terlebih dahulu dibersihkan dengan menggunakan *butil culfida* dan *tinner* dengan tujuan untuk mengangkat kotoran yang menempel pada *body*.
- *Stage 1 : Predegrease*  
*Body* dimasukkan pada bak berisi air panas (55-70°C) untuk menghilangkan oli dan garam pada *body*.
- *Stage 2 : Degrease*  
Pada tahap ini *body* direndam dengan air pam dan *ridoline* untuk memisahkan antara kotoran oli dan minyak pada *body*.
- *Stage 3 : Water Rinse*  
*Body* dibilas agar bersih dan tidak terjadi kontaminasi.
- *Stage 4 : Conditioner*  
Pada tahap ini dilakukan pelapisan primer pada *body*, digunakan *fixodin* untuk menghaluskan permukaan *body*.
- *Stage 5 : Phosphating*  
Pada tahap ini dilakukan pelapisan *phosphate* dengan tujuan agar *body* anti karat.
- *Stage 6 : Water Rinse 2*  
*Body* dibilas agar bersih dan tidak terjadi kontaminasi.
- *Stage 7 : Passivating*  
Pada stage ini tidak terjadi proses apapun, *body* di istirahatkan agar air bekas pembilasan yang ada pada *body* berkurang karena menetes.

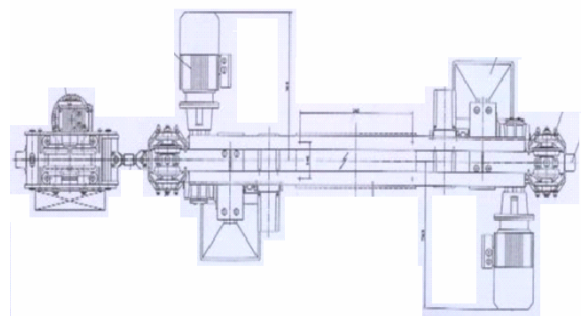
- *Stage 8 : DI Rinsing*  
Pada stage ini dilakukan pembersihan dengan *DI Water* (*Diozoned Water*) sebelum *body* di celupkan ke dalam cat.
- *Stage 9 : Electrocoat Deposition (ED)*  
Pada tahap ini dilakukan pengecatan warna dasar dengan cara mencelupkan *body* ke dalam cat. Sistem ini dikenal dengan istilah *cathodic electrocoating*, yaitu mengecat dengan bantuan polaritas listrik agar cat dapat menempel pada logam (*body*) sebagai polaritas negatif yang dicelupkan pada cat sebagai polaritas positif. Ketebalan cat ditentukan oleh besarnya tegangan listrik yang dikenakan pada sistem tersebut.

### A. Mesin Hoist Hanger

Mesin *hoist hanger* adalah mesin pemindah *body* mobil yang berjalan pada suatu rel. Mesin ini digunakan pada proses PT/ED di PT. Gaya Motor. *Body* diangkat oleh mesin ini (*hoist*) kemudian dipindah (*hanger*) ke tempat lain melalui rel. Sketsa mesin *hoist hanger* ditunjukkan oleh Gambar 9 dan Gambar 10.



Gambar 9 Sketsa mesin hoist hanger tampak samping.

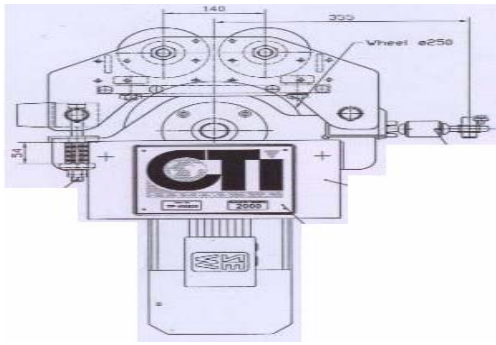


Gambar 10 Sketsa mesin hoist hanger tampak bawah.

Mesin *hoist hanger* terdiri dari dua *hoist*, *hanger*, tiga motor 3 fasa, rel, dan sumber tenaga. Untuk menggerakkan *hoist* maupun *hanger* digunakan motor 3 fasa, rel merupakan lintasan, sedangkan sumber tenaga sebagai penyedia sumber listrik.

### B. Prinsip Kerja Mesin Hoist Hanger

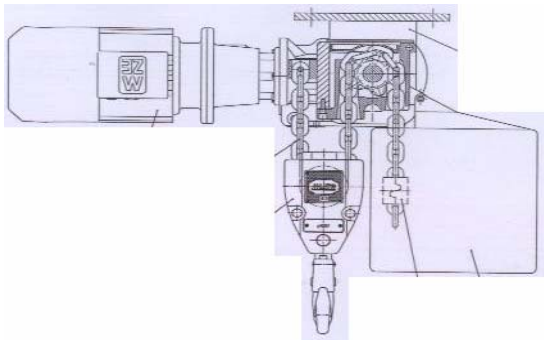
Mesin *hanger* adalah mesin yang berjalan pada suatu rel. Terdiri dari motor penggerak, roda penggerak, dan roda yang berjalan di rel. mesin *hanger* ditunjukkan oleh Gambar 11.



Gambar 11 Mesin Hanger

Mesin ini digerakkan oleh motor induksi 3 fasa yang terdiri dari line R, S, T, dan N. Untuk membalik keadaan atau agar *hanger* dapat berjalan mundur maka salah satu line ditukar penyambungannya.

Mesin *hoist* terdiri dari motor penggerak 3 fasa, rantai, dan pengait. Mesin *hoist* ditunjukkan oleh Gambar 12.



Gambar 12 Mesin hoist

Sama seperti mesin *hanger*, *hoist* dapat bergerak maju (turun) ataupun mundur (naik). Untuk membalik keadaan maka salah satu line ditukar penyambungannya. Pada *plant* yang ditinjau menggunakan dua *hoist*.

### Hasil dan Analisis

Pada dasarnya, semua perancangan program PLC adalah sama, yaitu dengan perancangan logika. Pemrograman pada PLC Omron menggunakan *software* ZEN Omron, yaitu dengan merancang diagram *ladder* yang merepresentasikan sistem *hoist hanger* secara keseluruhan. Metode perancangan yang digunakan adalah diagram *state* FSM Moore. Langkah awal dalam metode ini adalah inisialisasi I/O. Inisialisasi dilakukan untuk menentukan *input* dan *output*, sehingga sebelum membuat diagram *state*, maka perlu diketahui dulu *input* maupun *output* sistem.

*Input* : First scanning (FS)

Push button emergency (PB1)

Push button *hanger* maju (PB2)

Push button *hoist* depan turun (PB3)

Push button *hoist* belakang turun (PB4)

Push button *hoist* depan naik (PB5)

Push button *hoist* belakang naik (PB6)

Proximity switch (Px)

Timer on delay (tm0...4, tm0...3)

Limit switch (LS1...16)

*Output* : Relay motor *hanger* maju (Hngr)

Relay motor *hoist* depan turun (R1D)

Relay motor *hoist* depan naik (R1U)

Relay motor *hoist* belakang turun (R2D)

Relay motor *hoist* belakang naik (R2U)

Berdasarkan sistem pengendalian pada mesin *hoist hanger*, maka *state-state* yang relevan, yaitu:

S0 : Semua output mati.

S1 : relay motor untuk menggerakkan *hanger* aktif (*hanger* maju), *output* lainnya mati.

S2 : relay motor untuk menggerakkan *hoist* depan aktif (*hoist* depan turun), *output* lainnya mati.

S3 : relay motor untuk *hoist* depan dan untuk *hoist* belakang aktif (kedua *hoist* turun), *output* lainnya mati.

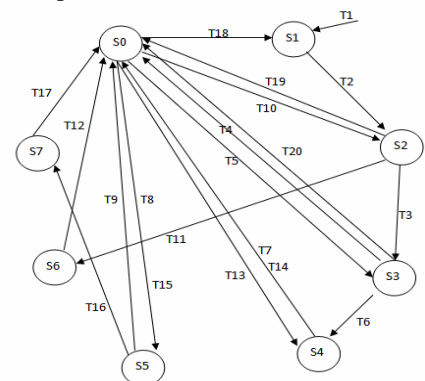
S4 : relay motor untuk *hoist* belakang aktif (*hoist* belakang turun), *output* lainnya mati.

S5 : relay motor untuk *hoist* depan aktif (*hoist* depan naik), *output* lainnya mati.

S6 : relay motor untuk *hoist* belakang aktif (*hoist* belakang naik), *output* lainnya mati.

S7 : relay motor untuk kedua *hoist* aktif (keduanya *hoist* naik), *output* lain mati.

Kemudian, *state-state* tersebut dibuat diagram *state* yang ditunjukkan pada Gambar 13.



Gambar 13 Diagram *state*.

Pada Gambar 13, *state* akan berpindah ke *state* 1 jika terjadi transisi T1 atau T8, sedangkan *state* akan berpindah ke *state* 2 jika terjadi taransisi T2 atau T10. Demikian seterusnya dengan proses perpindahan *state* berikutnya.

Dari *state* dan *output* yang ada, maka dapat dibuat tabel hubungan *state* dan *output* yang ditunjukkan oleh Tabel 1.



Tabel 1 Hubungan *state* dan *output*

<i>state</i>	Hngr	R1D	R1U	R2D	R2U
S0	0	0	0	0	0
S1	1	0	0	0	0
S2	0	1	0	0	0
S3	0	1	0	1	0
S4	0	0	0	1	0
S5	0	0	1	0	0
S6	0	0	0	0	1
S7	0	0	1	0	1

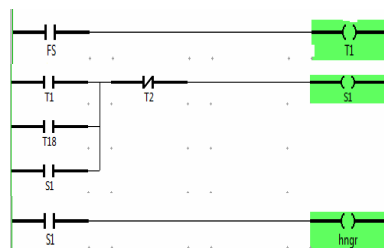
Perancangan diagram *ladder* dilakukan berdasarkan diagram *state*, tabel hubungan *output* dengan *state*, serta transisi *state*. persamaan-persamaan yang mendukung perancangan diagram *ladder* untuk *hanger* bergerak maju yaitu:

$T1 = FS$

$S1 = (S1 + T1 + T18). \sim T2$

$Hngr = S1$

Kemudian, dari persamaan-persamaan tersebut dibuat diagram *ladder* untuk *hanger* bergerak maju yang ditunjukkan pada Gambar 14.

Gambar 14 Diagram *ladder* untuk *hanger* maju

Pembacaan diagram *ladder* untuk *hanger* bergerak maju, yaitu pada Gambar 14, pada anak tangga pertama, penekanan *first scanning* (FS), akan mengakibatkan terjadi *energize* pada T1, sedangkan T1 sendiri digunakan untuk melakukan *energize* pada S1, sehingga menyebabkan sistem berpindah ke *state* 1 dan pada S1 yang mengalami *energize* akan menggerakkan *hanger*. *Hanger* akan bergerak jika T1, T18, atau S1 mengalami *energize*, sedangkan akan mati jika pada T2 yang terjadi *energize*.

### Kesimpulan

1. Penggunaan PLC dapat mengendalikan mesin hoist *hanger* baik dan mempermudah pengoperasian, sehingga dapat mengoptimalkan SDM yang ada.
2. Penggunaan metode diagram *state* mempermudah perancangan sistem kontrol dengan menggunakan PLC.
3. Penggunaan diagram *ladder* sangat menguntungkan karena jika sewaktu-waktu dibutuhkan dalam *maintenance* atau keperluan yang lain, dapat dengan mudah dianalisis.

### Daftar Pustaka

1. Bolton, William, 2002, *Programmable Logic Controller (PLC) sebuah pengantar*. Erlangga. Jakarta.
2. John W.W, 1999, *Programmable Logic Controller*, Fourth Edition, Prentice Hall, New Jersey.
3. M. Budiyo, A.Wijaya, *Pengenalan Dasar-Dasar PLC*, Penerbit Gaya Media, Yogyakarta, 2003.
4. Petruzella, Frank D., 2001, *Elektronik Industri*. Andi Offset. Yogyakarta.
5. Robert, L., 1991, *Industrial Electronics*, Fourth Edition, Mc. Graw-Hill Publishing Company, New York.
6. Setiawan, Iwan, 2006, *Programable Logic Controller (PLC) dan Teknik Perancangan Sistem Kontrol*. Yogyakarta. Andi Offset. 2006.
7. Tocci, Ronald J, 1991, *Digital Systems : Principles and Applications*, 5<sup>th</sup> Edition. Prentice –Hall. New Jersey.

